

Quelle précision le budget a-t-il achetée ?

La formule de taille d'échantillon, les quatre nombres qu'elle exige, et pourquoi la réponse pour une enquête en grappes vaut environ le double de celle des manuels. Plus sa lecture à l'envers, celle dont on a le plus souvent besoin.

Pierre Robentz Cassion

Leçon 3 sur 8

Analyse d'enquêtes, échantillonnage et pondération — Ce que le plan achète

Ce que couvre cette leçon

- La question est toujours posée trop tard
- La formule, et les quatre nombres qu'elle contient
- La lecture à l'envers
- Le nombre qui se négocie réellement
- Grappes contre ménages par grappe
- Écrivez le calcul
- La suite

La question est toujours posée trop tard

- La taille d'échantillon se décide dans une proposition, des mois avant toute analyse, et généralement par la personne qui rédige le budget.

La formule, et les quatre nombres qu'elle contient — Exemple

$$n = z^2 * p * (1 - p) / d^2 * deff$$

La formule, et les quatre nombres qu'elle contient

- z — le niveau de confiance. 1,96 pour 95 %. Presque jamais modifié.
- p — la proportion attendue. Vous ne la connaissez pas, c'est l'objet de l'enquête. Prenez le tour précédent, une. . .
- d — la précision voulue, en marge absolue. C'est le nombre réellement négocié, et celui que personne n'écrit.
- d_{eff} — l'effet de plan. La leçon suivante lui est entièrement consacrée ; pour dimensionner, prenez la valeur du. . .

La formule, et les quatre nombres qu'elle contient — En Python

```
import math

def sample_size(p, d, deff=1.0, z=1.96, response=1.0):
    n = z**2 * p * (1 - p) / d**2 * deff
    return math.ceil(n / response)

print(sample_size(0.30, 0.05))           # simple random
print(sample_size(0.30, 0.05, deff=2.4)) # cluster survey
print(sample_size(0.30, 0.03, deff=2.4)) # tighter precision
```

La formule, et les quatre nombres qu'elle contient — En R

```
sample_size <- function(p, d, deff = 1, z = 1.96, response = 1) {  
  ceiling(z^2 * p * (1 - p) / d^2 * deff / response)  
}
```

```
c(sample_size(0.30, 0.05),  
  sample_size(0.30, 0.05, deff = 2.4),  
  sample_size(0.30, 0.03, deff = 2.4))
```

La formule, et les quatre nombres qu'elle contient

p attendu	Précision	d _{eff}	Ménages nécessaires
30 %	±5 points	1,0	323
30 %	±5 points	2,4	775
30 %	±3 points	2,4	2 152

La formule, et les quatre nombres qu'elle contient

- La mise en grappes plus que double l'exigence — 323 devient 775 pour la même précision
- La précision coûte très cher — Passer de ± 5 à ± 3 points triple presque l'échantillon
- La non-réponse est un diviseur, pas un détail — Attendre 90 % de réponse signifie diviser par 0,9 — et cela s'applique. . .

```
def precision(n, p, deff=1.0, z=1.96):  
    return z * math.sqrt(p * (1 - p) / n * deff)  
  
print(f"+/- {precision(996, 0.291, deff=1.60):.1%}")  
print(f"+/- {precision(343, 0.464, deff=1.60):.1%}")    # rural remote alone
```

```
precision <- function(n, p, deff = 1, z = 1.96) z * sqrt(p * (1 - p) / n * deff)

c(overall = precision(996, 0.291, deff = 1.60),
  remote  = precision(343, 0.464, deff = 1.60))
```

- Faites ce calcul avant de promettre quoi que ce soit — Il dit quelles comparaisons l'enquête peut trancher et...

Le nombre qui se négocie réellement

- **Un seuil à franchir.** Si la question est de savoir si la MAG dépasse 15 % et que l'estimation est proche de 14 %, il...
- **Un changement à détecter.** Détecter une amélioration de cinq points entre deux tours exige environ *quatre fois*...
- **Une comparaison entre groupes.** Même problème — deux intervalles, tous deux à resserrer.

Le nombre qui se négocie réellement — En Python

```
# Detecting a difference between two groups of equal size
def sample_per_group(p1, p2, deff=1.0, power_z=0.84, z=1.96):
    pbar = (p1 + p2) / 2
    n = (z + power_z) ** 2 * 2 * pbar * (1 - pbar) / (p1 - p2) ** 2
    return math.ceil(n * deff)

print(sample_per_group(0.30, 0.25, deff=2.4))
```

Le nombre qui se négocie réellement — En R

```
sample_per_group <- function(p1, p2, deff = 1) {  
  pbar <- (p1 + p2) / 2  
  ceiling(((1.96 + 0.84)^2 * 2 * pbar * (1 - pbar) / (p1 - p2)^2 * deff)  
}  
  
sample_per_group(0.30, 0.25, deff = 2.4)
```

Grappes contre ménages par grappe — En Python

```
for m in [8, 14, 20, 30]:  
    deff = 1 + (m - 1) * 0.11  
    clusters = math.ceil(775 * (deff / 2.4) / m)  
    print(f"{m:>3} households x {clusters:>3} clusters -> deff {deff:.2f}")
```

Grappes contre ménages par grappe — En R

```
for (m in c(8, 14, 20, 30)) {  
  deff <- 1 + (m - 1) * 0.11  
  cat(sprintf("%3d households, deff %.2f\n", m, deff))  
}
```

Écrivez le calcul — Exemple

Indicator	Household food insecurity prevalence
Expected p	0.30 (2023 round, same instrument)
Precision	+/- 5 percentage points, 95% confidence
Design effect	2.0 (2023 round measured 1.9; rounded up)
Expected response	92%
Households	$323 * 2.0 / 0.92 = 703$, rounded to 25 clusters x 14 = 350 per stratum, 1,050 in total across three strata
Rationale	Equal allocation, because each stratum must support its own estimate. National figures require weighting.

- Deux des quatre entrées étaient l'effet de plan, resté jusqu'ici un provisoire.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/survey-analysis-sampling/
survey-03-sample-size](https://data-analysis.cassion.dev/fr/cours/survey-analysis-sampling/survey-03-sample-size)